We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

2

Intelligent Agent Technology in Modern Production and Trade Management

Serge Parshutin, Arnis Kirshners
Institute of Information Technology, Riga Technical University
Riga, Latvia

1. Introduction

The technological progress is constantly evolving and each year presents new growth opportunities for different people and companies. Modern production and trade management are processing terabytes of gathered statistical data to gain useful information for different management tasks. Most common tasks that have arisen for solving are forecasting the demand for a new product, knowing only the product descriptive data and creating a production plan for a product on different product life cycle phases. As the amount of data grows, it becomes impossible to analyse it without using modern intelligent technologies.
Trade management is strictly connected with sales forecasting. Achieving accurate sales forecasting will always remain on the top of the most important tasks. Due to that different solutions are introduced each year in order to overcome the efficiency of the traditional sales forecasting methods. Let us discuss several possible solutions recently introduced. A sales forecasting model based on clustering methods and fuzzy neural networks was proposed by (Chang et al., 2009). The model groups demand time series applying clustering algorithms, then a fuzzy neural network is created for each cluster and used to forecast sales of a new product. Another interesting solution for sales forecasting was proposed by (Ni & Fan, 2011). The model combines the Auto-Regressive trees and artificial neural networks to dynamically forecast a demand for a new product, using historical data. Both models apply modern intelligent information technologies and, as authors state, are more efficient and accurate comparing to traditional forecasting methods. Nevertheless, the models preserve that a product is already introduced on the market and some sales data are already available. This rises a question that may be important for retailers, supplying market with items, but not producing them personally. The question may be stated as "How to forecast a demand for a product before purchasing it for resale?". A system, capable of making such forecasts, will bring to retailers a useful and valuable information, enabling them to create a list of stock products before any investments are made. A possible solution to such task was proposed by (Thomassey & Fioraliso, 2006). The authors proposed a hybrid sales forecasting system based on a combination of clustering and classification methods. The system applies clustering methods to gain product demand profiles from historical data, then, using product descriptive data, a classifier is built and used to forecast a sales curve for a new product, for which only descriptive data are available. The system was tested on textile item sales and, as authors state, the results were relatively good. Authors continued their research and in (Thomassey & Happiette, 2007) a new system based on idea of combining clustering and classification methods was introduced. Authors used Artificial Neural Networks technologies to cluster

and classify sales time series. As they state, the proposed system increases accuracy of mid-term forecasting in comparison with the mean sales profile predictor.

The production management itself combines several tasks of forecasting and planning, one of which is a product life cycle (PLC) management. A product life cycle is a time of product existence on the market and can be viewed as a sequence of known phases with bounds. Each PLC phase differs by sales profiles, due to what different production planning strategies are used and what is more important - different advertising strategies are applied. Traditionally a product life cycle has four phases, namely "Growth", "Introduction", "Maturity" and "End-of-Life". The models with a larger number of phases are also possible and are used (Aitken et al., 2003). For products with a significantly long PLC, growth and introduction phases can be merged into one phase - "Introduction". This PLC phase imply high investments in advertising, as potential customers need to know about the new product. This gives rise to the task of defining bounds for the PLC phases, which to be one of the important tasks in the PLC management. Let us discuss several possible solutions introduced for this task. Most of the solutions concentrate around the idea of forecasting the sales curve for a product and then defining in which particular PLC phase product currently is. Often the *Bass* diffusion model or its modifications are used to forecast sales, as it was proposed by (Chien et al., 2010). Authors present a demand forecasting framework for accurately forecasting product demand, thus providing valuable information to assist managers in decision-making regarding capacity planning. The framework is based on the diffusion model and, as authors state, has a practical viability to forecast demands and provide valuable forecasting information for supporting capacity planning decisions and manufacturing strategies. Other researchers (Venkatesan & Kumar, 2002) applied a genetic algorithm to estimate the *Bass* diffusion model and forecast sales for products, when only a few data points are available. Both solutions make accurate forecasts in the scope of the tasks solved, but still give an answer to what the demand will be, but not to when the PLC phase will be changed. To answer that, the forecasted sales curves need to be analysed by a manager, who will set PLC phase bounds, using his personal experience. Due to that, the efficiency of the discussed models may fall as the amount of monitored products growths.

The present research is oriented towards bringing intelligent agent technology for supporting manager's decisions in such tasks as forecasting demand for a new product and defining the end of the introduction/beginning of the maturity PLC phase for the monitored products. The chapter proposes two multi-agent systems designed directly for solving the tasks defined. Intelligent agent technology was chosen as it gives a new modern look at the forecasting and planning tasks and brings new possibilities in designing solutions to support modern production and trade management.

The conceptual and functional aspects of the proposed systems are revealed in Sections 2 and 3, but first, some agent terminology needs to be defined to eliminate possible misunderstandings. There are no unified definitions of what an agent is (Wooldridge, 2005). One of the reasons for that is that agents are mostly defined as a task specific concept. Nevertheless, some definitions are useful (Weiss, 1999; Wooldridge, 2005) and also applicable to present research:

- Agent is a computer program that exists in some environment and is capable of performing an autonomous activity in order to gain its desired objectives.
- Intelligent agent is an agent capable of learning, capable of mining and storing knowledge about the environment it is situated in.
- Multi-agent system contains agents and intelligent agents capable of interaction through an information exchange.

- Agent community is an indivisible unit of agents with similar attributes, abilities and behaviour, aimed at gaining the desired objectives. It differs from a Multi-agent system in that agents are not interacting with one another.

The proposed definitions are task specific and are oriented to making the ideas proposed in the multi-agent systems simple and clear.

2. Agent Technology in PLC Management

The product life cycle management, as it was stated in the introduction, is an important objective for company growth, as most of the decisions taken are dependent on the current demand. A proper product life cycle management strategy gives opportunities for precise resource planning, lessening expanses and negative profit.

The present section reveals the concepts of the proposed PLC management support multi-agent system, and describes both structural and functional aspects of it. An analysis of the obtained experimental results concludes this section.

The main task of the proposed system is to forecast a period - transition point, when a product will change an introduction phase to maturity phase. A specific feature of the system is that the forecast is made having only the first several data points, and a scalar value - period, is forecasted. From the viewpoint of Data Mining this task is a classification task and formally can be stated as follows. Assume that the available sales data is a dataset $D = \{d_1, \ldots, d_i, \ldots, d_n\}$, where each record $d = \{x_1, \ldots, x_j, \ldots, x_l\}$ is a discrete time series with l periods, representing a sales profile of a specific product during the defined phase of a product life cycle. The length of a time series l is not a constant value. It may take a finite number of values defined by the set L, $l \in L = \{l_1, \ldots, l_h, \ldots, l_s\}$ and may vary from record to record.

Each record $d_i \in D$ has a specific marker p containing the number of a period after which a PLC phase, represented by record d_i, was finished. Marker p may take a finite number of values defined by the set P, $p \in P = \{p_1, \ldots, p_k, \ldots, p_m\}$. Having such assumptions the forecasting of a transition point for a specific new product $d' \notin D$ should start with building a model of an implication between datasets D and P, $f : D \to P$, representing relations between sales profiles and the value of a specified marker. As the model is built, it may be applied to forecast a transition point for a new record, when it reaches the minimal length $l_{min} \in L$.

2.1 Structure of the proposed system

The proposed product life cycle management support multi-agent system is displayed in Figure 1. The system contains three interacting agents, namely Data Management Agent, Data Mining Agent and Decision Support Agent. The Data Management Agent was included as the system should be capable of processing a large amount of sales data. This condition asks for an agent capable of autonomously performing general and specific operations with data, one of which is pre-processing. As it was stated before, first, the model of the relations between the sales profiles and a target attribute should be made. The Data Mining Agent implements an intelligent data processing methods and is capable of autonomously creating a knowledge base with desired relations. The application of the knowledge base to the forecasting of a transition point for a new product is designated to the Decision Support Agent.

2.1.1 Data Management Agent

The Data Management Agent is an agent as it uses a predefined logic and cannot change it over time. The agent handles several data management tasks, including data cleaning, normalization and preparing datasets. The environment the Data Management Agent is

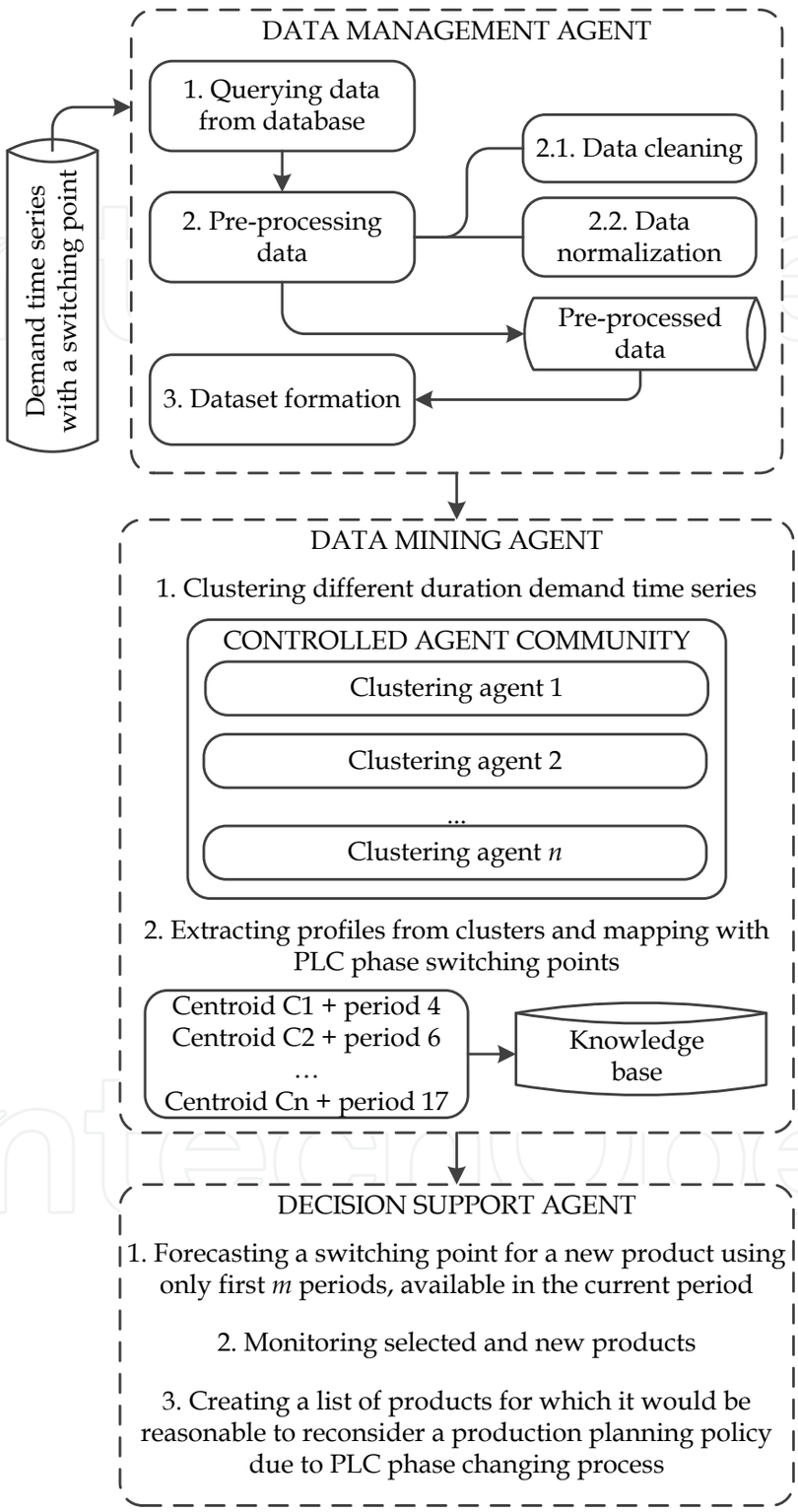


Fig. 1. PLC Management Support Multi-Agent System

situated in is a raw data repository. The environment affects the agent with new data, forcing the Data Management Agent to perform the data pre-processing actions. By performing those actions, the agent affects the environment by changing the raw data to the pre-processed data. The agent functioning algorithm, displayed in Figure 1, contains three main processes - first get data from a database, then pre-process it and prepare datasets for system training and testing. The first two processes are performed in the autonomous mode, the actions in the third process are performed reactively by responding to the Data Mining Agent's requests.
The data repository contains data of the unique structure that may not fit the desired data format. The processes in the first block of an Data Management Agent algorithm return the set of sales data, obtained during a specified product life cycle phase, as a time series, summing sales by each period (day, week, month), which is defined by the user. For each sales time series a value of a target attribute (transition point) is supplied, excluding the new data, for what the information on the transition point is not available yet. The next step is pre-processing, it becomes necessary as the raw data may contain noise, missing or conflicting data. The defined task speaks for performing several steps of data pre-processing:

- **Excluding the outliers.** The definition of an outlier is dynamic and changes according to the desired tasks. This makes the step of outlier exclusion highly task specific. The present research foresees the following actions to be completed. The time series with the number of periods (length) less than the user defined minimum l_{min} or with missing values to be excluded from the raw dataset.
- **Data normalization.** This step is performed in order to bring sales time series to one level, lessening possible domination occurrences. One of the normalization methods that suites the sales time series normalization is the *Z-score* normalization with standard deviation method. To calculate the normalized value $x_i^{'}$ for i-th period of a time series X, Equation 1 is used, where $\overline{X}$ is the average of an X.

$$x_i^{'} = \frac{x_i - \overline{X}}{s_x} \tag{1}$$

  To calculate the standard deviation s_x, Equation 2 is used. As the number of periods in the normalized sales time series is not large, it is recommended to use the $(n-1)$ as a denominator in order to obtain an unbiased standard deviation. As an option, the standard deviation can be replaced with a mean absolute deviation.

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2}$$

- **Imitation of a data flow.** The sales data become available over time and this moment should be taken into account in order to obtain an accurate transition point forecast. This step is very important as the proposed system will be limited to forecast a binary result - either the transition is occurred or not, if the data flow imitation will not be performed. The imitation of the data flow process ensures that the proposed system will be able to forecast transition points in different time periods having only first l_{min} periods available.

2.1.2 Data Mining Agent

The Data Mining Agent (see Fig. 1) is an intelligent agent that performs an intellectual data processing with data mining methods. The environment the agent is situated in is the datasets created by the Data Management Agent. This environment is highly dynamic as it changes

each time a new data is obtained. The agent itself may induce changes in the environment by requesting the Data Management Agent to produce new datasets. The Data Mining Agent changes the environment by creating the knowledge base, that contains relations between the sales time series and transition point values. The proposed structure of a Data Mining Agent contains two main steps - first clustering the time series with different duration and then extracting profiles from clusters and mapping profiles with PLC phase switching points.
The time series clustering is performed by the clustering agents in the agent community controlled by the Data Mining Agent (see Fig. 1). Each clustering agent implements a clustering algorithm and processes a defined part of the input data. The number of the clustering agents as well as the part of dataset it will process, is defined by the parameter Q called the load of a clustering agent. It is an integer parameter containing a number of different time series lengths $l_i \in L$ each clustering agent can handle. The set L contains all possible lengths of the time series in the input data, as it was defined in the task statement at the beginning of this section. For example, assume that $l_{min} = 4$ periods and $Q = 3$, then the first clustering agent will handle the part of an input data with 4, 5 and 6 periods long time series. The load of a clustering agent is defined by the user and is also used for calculating n_{ca} - the number of the clustering agents in the Controlled Agent Community (CAC). Equation 3 is used for obtaining the value of n_{ca}.

$$n_{ca} = Roundup\left(\frac{|L|}{Q}\right) \tag{3}$$

The input data is distributed among n_{ca} clustering agents $ca_i \in CAC$ according to the defined value of Q. Given a uniform clustering agent load distribution, Equation 4 can be used for splitting the input data among clustering agents.

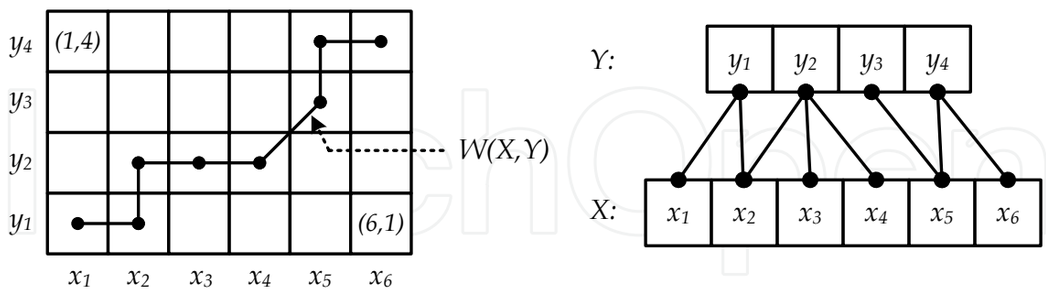
$$\begin{cases} i = 1, & l_{i,min} = l_{min} \\ i > 1, & l_{i,min} = l_{i-1,max} + 1 \end{cases} \qquad l_{i,max} = l_{i,min} + Q - 1 \tag{4}$$

where the l_{min} and l_{max} are the bounds for the number of periods in the time series the clustering agent ca_i will process. One of the proposed system objectives is to support clustering of time series with different number of periods. A number of clustering methods exist, but majority use an Euclidean distance to calculate the similarity of two objects. The simple Euclidean distance will not allow a clustering agent to process time series with different number of periods, but still may be applied while $Q = 1$. Two clustering methods - *Self Organising Maps* (SOM) (Kohonen, 2001) and *Gravitational Clustering algorithm* (GC) (Gomez et al., 2003; Wright, 1977), were chosen to support the data mining process. To gain an ability to cluster time series with different duration we suggest using different distance measures in each of the methods. The proposed system implements two different distance measures - *Dynamic Time Warping* (DTW) (Keogh & Pazzani, 2001; Salvador & Chan, 2007) and a modified Euclidean distance *MEuclidean*.

2.1.2.1 Distance measures

The *DTW* creates a warping path $W(X, Y)$ between two time series X and Y and uses it to calculate the distance, which occurs as follows. Let X and Y have durations equal to n and m periods, respectively. First a distance matrix $n \times m$ is created, where each cell (i, j) contains distance $d(x_i, y_j)$ between two points x_i and y_j (Keogh & Pazzani, 2001). The warping path $W(X, Y)$ is a sequence of the steps w_k in the distance matrix starting with a cell $(1, 1)$ and

ending in (n, m). Each time performing a new step w_k, a direction with minimal distance is chosen. Other strategies for choosing the next step are possible and widely used (Salvador & Chan, 2007). An example of a warping path is graphically displayed in Figure 2.



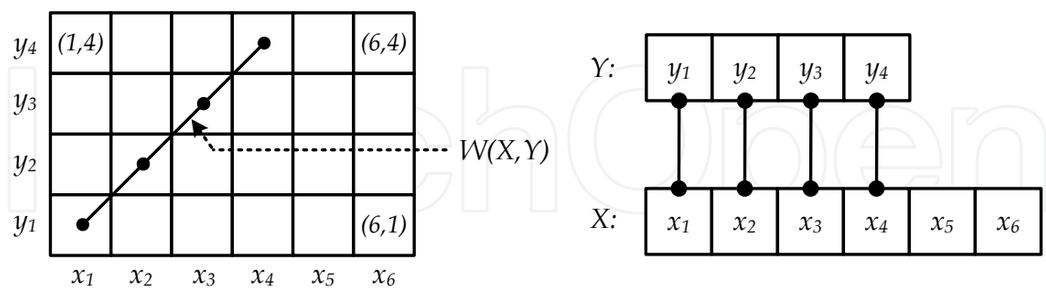
(a) Warping path in the distance matrix (b) A warping map M

Fig. 2. Example of a *DTW* warping path $W(X, Y)$

The total path distance $d(X, Y)$ between two time series X and Y using the Dynamic Time Warping is calculated as shown in Equation 5. The denominator K representing the total number of steps w_k in the warping path $W(X, Y)$ is used in order to calculate the distance proportionally for one step w_k, as the number of steps may be different in different warping paths.

$$d(X, Y) = \frac{\sqrt{\sum_{k=1}^{K} w_k^2}}{K} \tag{5}$$

The second proposed distance measure is *MEuclidean* - Modified Euclidean. The concept of this measure is to calculate the distance $d(X, Y)$ between two time series $X = x_1, \ldots, x_i, \ldots, x_n$ and $Y = y_1, \ldots, y_j, \ldots, y_m$ only by first z periods of each time series, where $z = min\{n, m\}$. Speaking in the terminology of *DTW*, the warping path $W(X, Y)$ in the case of *MEuclidean* will always begin in the cell $(1, 1)$ and will always end in the cell (z, z). The example of a warping path for a distance measure *MEuclidean* is displayed in Figure 3.



(a) Warping path in the distance matrix (b) A warping map M

Fig. 3. Example of a *MEuclidean* warping path $W(X, Y)$

The total path distance $d(X, Y)$ between two time series X and Y using the *MEuclidean* distance measure is calculated using Equation 6.

$$d(X, Y) = \sqrt{\sum_{i=1}^{z} (x_i - y_i)^2} \tag{6}$$

2.1.2.2 Clustering algorithms

The *SOM* model introduced by T.Kohonen is one of the vector-coding algorithms, whose main objective is to transform an input data into a discrete map of neurons. The *SOM* algorithm contains four main steps - network initialization, neuron competition, neuron cooperation and synoptic adaptation (Kohonen, 2001). Executing the initialization,weights of each neuron are assigned random values small enough to eliminate any possible organization in the network at the beginning of the system training.
During the neuron competition step, for each time series in the input dataset the winning neuron, as the one with most similar vector of weights, is defined. This is where the proposed distance measures are applied to find the winning neuron, which will be set as the centre of the topological neighbourhood in order to perform the neuron cooperation step. The cooperating neurons are neurons that are stimulated by the activity in the winning neuron. Executing the cooperation step each neuron in the *SOM* is assigned a neighbourhood level, using the neighbourhood function $h_{j,i(X)}$, where $i(X)$ is the index of a winning neuron for a time series X and j is an index of a cooperating neuron. The neighbourhood level is also affected by the chosen neural network topology. The classical *SOM* algorithm uses Equation 7 for synoptic adaptation of weights.

$$W_j(n+1) = W_j(n) + \eta(n) \cdot h_{j,i(X)}(n) \cdot (X - W_j(n)) \,, \tag{7}$$

where n is the iteration number; $W(n)$ - vector of weights of a j-th neuron; η is a learning coefficient.
The learning coefficient decreases over time, but remains higher than a defined minimal value η_{min}. This can be reached by using Equation 8 (Kohonen, 2001), where τ_2 is a time constant, which can be assigned total iterations number. The value of a topological neighbourhood function $h_{j,i(X)}(n)$ at the moment n is calculated by Equation 9.

$$\eta(n) = \eta_0 \cdot exp\left(-\frac{n}{\tau_2}\right) \tag{8}$$

$$h_{j,i(X)}(n) = exp\left(-\frac{d^2_{j,i(X)}}{2\sigma^2(n)}\right), \tag{9}$$

where d is the lateral distance between the j-th neuron and the winning neuron; parameter σ defines the width of a topological neighbourhood function.
In Subsection 2.1.2.1 it was shown that both of the proposed distance measures can produce a warping path in order to compare two time series. If a distance measure *MEuclidean* is used, then Equation 7 is modified and Equation 10 is used for synoptic adaptation of neuron weights.

$$\forall w_k \in W_j,\ k \leq l_{min} :\ w_k(n+1) = w_k(n) + \eta(n) \cdot h_{j,i(X)}(n) \cdot (x_k - w_k(n)) \,, \tag{10}$$

where W_j is the weight vector of a j-th neuron and $l_{min} = min(l_X, l_W)$.
Using distance measure *MEucliden* two cases during weight adaptation are possible:

1. The length l_W of a weight vector W_j of the j-th neuron is less or equal to the l_X - the length of a time series X. In this case all weights of the j-th neuron will be changed according to the warping map M, example of which is shown in Figure 3.b.

2. The length l_W of a weight vector W_j of the j-th neuron is greater than the l_X - the length of a time series X. In this case only the first l_X weights $w_k \in W_j$ will be adopted according to the warping map M. The rest of neuron weights will remain unchanged.

If the distance measure *DTW* is used, then cases are possible, when one weight w_k should be adopted to several values of a time series X. An example of this case is shown in Figure 2.b. Assuming that Y is a weight vector, the weights y_1, y_2 and y_4 must be adopted to two or three different values at one time. In this case each of the mentioned weights will be adopted to the average of all time series X values it must be adopted to, according to the warping map M.
This concludes the modifications of a *SOM* algorithm, the rest part of the subsection is devoted to the second chosen clustering algorithm - gravitational clustering algorithm. This algorithm is an unsupervised clustering algorithm (Gomez et al., 2003; Wright, 1977). Each clustering agent in the *CAC* represents a multidimensional space, where each time series from the input dataset is represented as an object in the n-dimensional space, where each object can be moved by the gravitational force of the other objects. If two or more objects are close enough, that is distance d between them is less or equal to the d_{max}, then those objects are merged into the cumulative object. By default, masses of all non-cumulative objects are equal to 1. Depending on user choice, masses of the merged objects can be summed or remain equal to 1. A movement of an object X, induced by the gravitational force of an object Y at the time moment t, can be calculated by Equation 11.

$$X(t+\Delta t) = X(t) + v(t)\cdot\Delta t + \vec{d}(t)\cdot\frac{G\cdot m_Y\cdot\Delta t^2}{2\cdot\left\|\vec{d}(t)\right\|^3}\ , \tag{11}$$

where $v(t)$ is the object velocity; G - gravitational constant, which can also be set by the user; m_Y - the mass of an object Y; vector $\vec{d}(t)$ represents the direction of a gravity force, induced by the object Y and is calculated as $\vec{d}(t) = X(t) - Y(t)$.
Setting velocity $v(t)$ as a null vector and having $\Delta t = 1$, Equation 11 can be simplified as shown in Equation 12.

$$X(t+\Delta t) = X(t) + \vec{d}(t)\cdot\frac{G\cdot m_Y}{2\cdot\left\|\vec{d}(t)\right\|^3}\ , \tag{12}$$

The *GC* uses the *Euclidean* distance to measure spacing between to objects. The support of clustering of time series with different number of periods can be gained by using the distance measures, proposed in Subsection 2.1.2.1, but the first of the gravitational clustering algorithm concepts should be redefined. In the classical algorithm this concept is defined as follows: "In the n-dimensional space all objects have values in all n dimensions". The proposed redefined version is "In the n-dimensional space can exist m-dimensional objects, where $m \leq n$". This means that in a 10-dimensional space can exist an object with four dimensions, which will be present only in the first four dimensions.
In the case of using the distance measure *MEuclidean* two possible cases can occur while calculating the induced gravitational force:

1. The number of the object's X dimensions n_X is less or equal to the number of the inducting object's Y dimensions n_Y. In this case the gravitational force is calculated for all dimensions of an object X and it is moved according to the calculated gravitational force.

2. The number of the object's X dimensions n_X is greater than the number of the inducting object's Y dimensions n_Y. In this case the gravitational force for X is calculated only for the first n_Y dimensions of the X. The object X is moved only in those dimensions for which the gravitational force was calculated.

Using the dynamic time warping, an m-dimensional object X will always be moved in all m dimensions. Situations, when for some dimension x_i a gravitational force should be calculated using several dimensions of an object Y, may take place. This is similar to the case with *SOM* when a single weight should be adopted to multiple time series values. Having a such situation, the gravitational force, induced by the object Y, is calculated using an average from all y_j dimensions inducing gravitational force to the object's X dimension x_i.

2.1.2.3 Creating a knowledge base

The knowledge base contains a model, representing the connections between sales profiles and the transition points, induced from the input data. The knowledge base is level-structured, the number of levels coincides with the number of the clustering agents in the clustering agent community. Each level contains clusters from a corresponding clustering agent. Let us define the meaning of a cluster for each of a clustering algorithm described in the previous subsection.

The cluster extraction in the *SOM* algorithm begins after network organisation and convergence processes are finished, and contains next three steps:

1. Each record X from a training set is sent to the system.
2. The winning neuron for each record X is found. For each winning neuron a transition point statistics S_j is collected, containing a transition point value and frequency of it's appearing in this neuron, which is used as a rank of a transition point.
3. The cluster is a neuron that at least once became a winning neuron during the cluster extraction process. The weight vector of a neuron is taken as a centroid of a cluster.

For the gravitational clustering algorithm a cluster is defined as follows: "The cluster is a cumulative object having at least two objects merged". Using the merged objects, a transition point statistics S_j is collected for each cluster c_j. Each transition point, included in S_j, is assigned a rank as the frequency of it's appearing in the cluster c_j. The centroid of a cluster is set as a position of a cumulative object in the clustering space at the moment of cluster extraction.

2.1.3 Decision Support Agent

The Decision Support Agent is an agent that forecasts a transition point value for a new data, monitors selected and new products and creates a list of the products, for which it would be reasonable to reconsider a production planning and advertising policies. The DSA can perform its actions either by receiving a request from a user, or in autonomous mode, with defined interval of time (at the end of each period) reporting the decision analysis results.

The transition point for a new data is forecasted by using the data from a cluster, best-matching the new time series. The best-matching cluster is obtained by sending an appropriate request to the Data Mining Agent, which finds the closest cluster in the knowledge base. The transition point with a higher rank is taken as the value that will be forecasted for the new time series. If several transition points have equal rank, then the minimal transition point is chosen.

The list of the noteworthy products (LNP) is created by applying this strategy:

1. If $l < p$ and $p - l > \theta$ Then: Product remains monitored and is not included in the LNP;

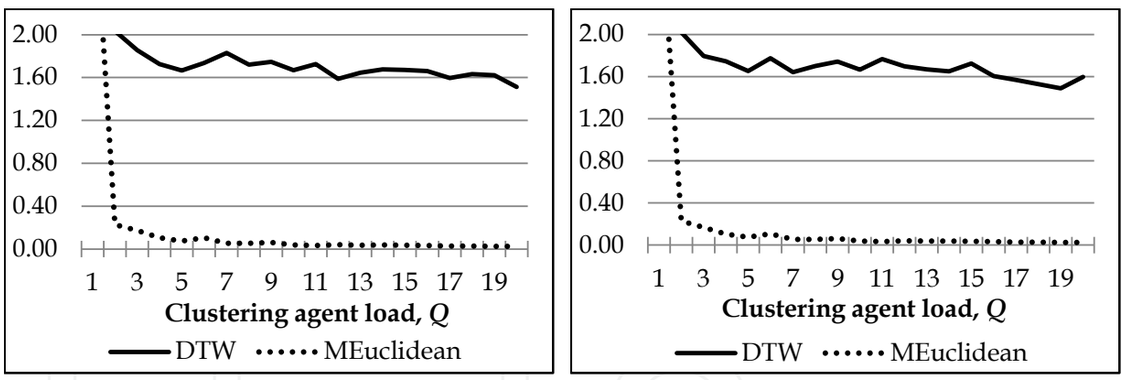
2. If $l < p$ and $p - l \leq \theta$ Then: Product is included in the LNP;
3. If $l \geq p$ Then: Product is included in the LNP.

where l is the duration of the demand time series in periods; p - a forecasted transition point; and variable θ stores the minimal threshold of interest for either including the product in the list of the noteworthy products or not.

2.2 Experimental results

The proposed product life cycle management support system was trained and tested on a dataset, received within the international project *ECLIPS*. The dataset contains sales profiles of different products during the PLC introduction phase. For each record in the dataset a transition point value is supplied. The length of a time series in the dataset varies from 4 to 23 periods, giving 20 possible values for a clustering agent load Q parameter, $Q = \{1, 2, \ldots, q_i, \ldots, 20\}$. The dataset was pre-processed according to the steps described in Subsection 2.1.1 and normalized by the *Z-score* normalization method.

The precision of the proposed system is measured, using the *Mean Absolute Error*, measured in periods. To measure the precision of the system, first the learning error for all combinations of the proposed clustering methods and distance measures was measured. The system with gravitational clustering algorithm was tested with both options for calculating the mass of a cumulative object after merging two or more other objects. The obtained results are displayed in Figure 4.



(a) Masses are summed (b) Object mass remains equal to 1

Fig. 4. Learning error (MAE) in periods for the system with gravitational clustering algorithm

As may be seen from Figure 4, the results, received using the distance measure *MEuclidean*, are more precise than those obtained with the *DTW*. The *MEuclidean* dominates *DTW* within all possible Q values, and that is the reason to exclude the *DTW* distance measure from further experiments with testing the system with gravitational clustering algorithm. As can also be seen, the results, obtained using *MEuclidean* in Figures 4.a and 4.b, are close to be equal. This indicates that both strategies for calculating the mass of a cumulative object are equal. However the strategy, where masses are summed after merging the objects, has a weakness. Merging the objects results in the growth of a mass of a cumulative object, consequently increasing the gravity force of that object. This could lead to the situation, when merging of objects will be affected more by the masses of an objects either by the closeness of those objects.

This could be the reason for that some potentially good combinations of clusters will not appear during the clustering process. Due to that, the strategy with summing objects masses will not be used in further experiments while using the gravitational clustering algorithm.
The transition point forecast error (using *MEuclidean*) decreases while the clustering agent load increases. This is the reason for performing further experiments by using only five values of Q that returned the smallest learning error. Those values of a clustering agent load are 16, 17, 18, 19 and 20, that is $Q = \{16, 17, 18, 19, 20\}$.
The system with *SOM* algorithm was tested using three neural network topologies - quadratic topology with eight neighbours in the first lateral level, cross-type topology with four neighbours and linear topology with two neighbours. The neural network contained 100 neurons, organized as a matrix 10×10 neurons for quadratic and cross-type topologies and as a one-dimensional line for the linear topology. The obtained results with learning error for each topology and distance measure *MEuclidean* are displayed in Figure 5.

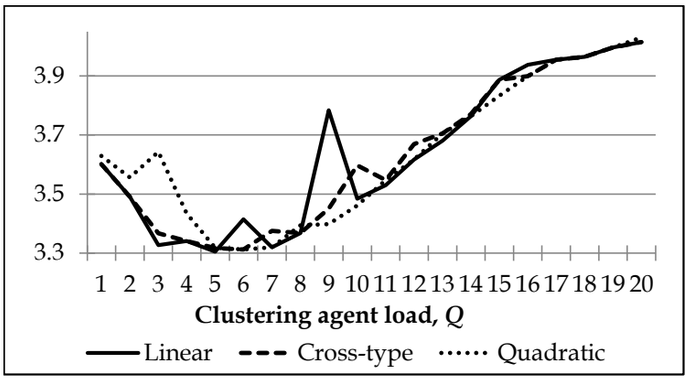


Fig. 5. Learning error (MAE) in periods for the system with *SOM* algorithm, using *MEuclidean*

The *SOM* algorithm returned a much larger learning error than the gravitational clustering algorithm. Due to that, only three values of Q, $Q = \{5, 6, 7\}$ will be used in further experiments. Table 1 contains the learning error obtained with *SOM* algorithm while using *DTW* and selected Q values.

	$Q = 5$	$Q = 6$	$Q = 7$
Linear	3.381	3.320	3.323
Cross-type	3.366	3.317	3.323
Quadratic	3.354	3.317	3.323

Table 1. Learning error (MAE) in periods for *SOM* algorithm, using *DTW*

Comparing the efficiency of the chosen topologies (see Figure 5) it may be concluded that none of three topologies dominates the others two. For further experiments with *SOM* algorithm all three topologies were chosen.
The learning error evaluation was used to lessen the number of experiments needed for system strong evaluation. To evaluate systems the 10-fold crossvalidation method was applied. The proposed system with gravitational clustering was tested, applying only the *MEuclidean* distance measure and using the five values of the clustering agent load, selected at the beginning of this section, while evaluating the system learning error. Table 2 shows the obtained system testing results, each cell contains a 10-fold average *Mean Absolute Error* in periods. Figure 6 displays the data from Table 2.

	Q = 16	*Q = 17*	*Q = 18*	*Q = 19*	*Q = 20*
MAE	2.021	**1.995**	2.026	2.019	2.010

Table 2. Mean absolute error in periods for the system with *GC* algorithm

As can be seen, the error remains at the level of two periods, the best result was gained with the clustering agent load $Q = 17$ and is 1.995 periods.

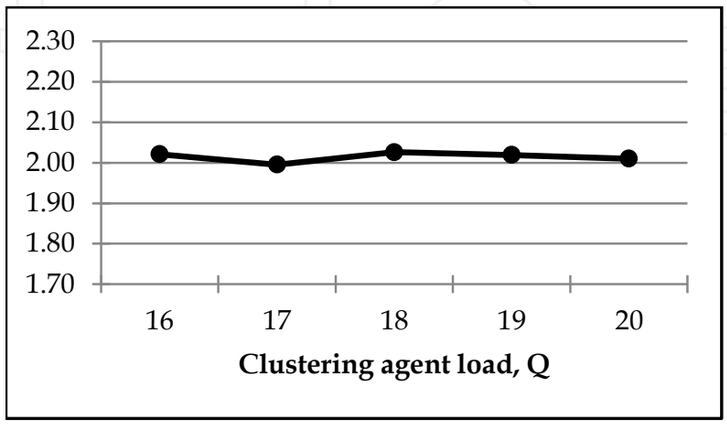


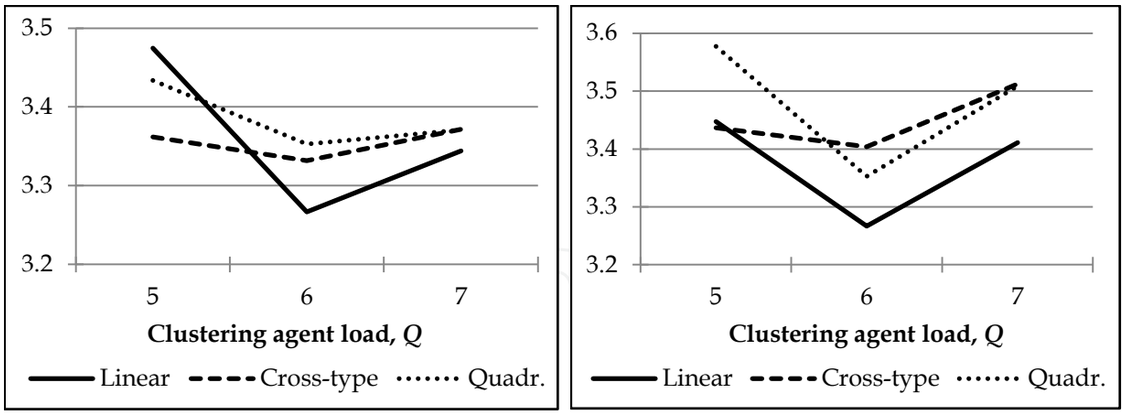
Fig. 6. MAE in periods for system with gravitational clustering, using *MEuclidean*

The same testing method and data were used for testing the system with *SOM* algorithm. While testing the system, all three neural network topologies - linear, cross-type and quadratic, were applied, combining with two proposed distance measures (see Subsection 2.1.2.1). The system with *SOM* was tested by using only three clustering agent load values - $Q = \{5, 6, 7\}$, selected while evaluating the learning error. Table 3 summarises the obtained testing results and Figure 7 displays them.

Q	**Linear topology**		**Cross-type topology**		**Quadratic topology**	
	DTW	*MEuclidean*	*DTW*	*MEuclidean*	*DTW*	*MEuclidean*
5	3.475	3.447	3.362	3.437	3.434	3.578
6	**3.267**	**3.267**	**3.332**	3.404	**3.353**	**3.353**
7	3.344	3.411	3.372	3.513	3.371	3.509

Table 3. Mean absolute error in periods for the system with *SOM* algorithm

The best results for all three topologies were obtained while testing the system with clustering agent load $Q = 6$ and remains equal about 3.3 periods. That supports the conclusion, stated while evaluating the learning error that none of three topologies dominates other two. The best result was obtained while using the linear topology and is equal to 3.267 periods. Comparing the efficiency of applied distance measures we may conclude that for the proposed system with *SOM* algorithm it is not crucial whether to choose *DTW* or *MEuclidean* distance measure, as the *DTW* returned better result only once, using the cross-type topology. For other topologies both distance measures returned equal results.



(a) Using distance measure *DTW* (b) Using distance measure *MEuclidean*

Fig. 7. Test error in periods for system with *SOM* algorithm

2.3 Results analysis and summary

Analysing the obtained results it may be concluded that for the task of forecasting the period when the product will change its PLC phase, the efficiency of the hierarchical clustering algorithms, a part of which is the gravitational clustering algorithm, comparing to the self-organising maps. Comparing the learning and test errors for both of clustering algorithms, it can be seen that the *GC* test error (1.995 periods) is about 10 times greater than the learning error (0.2 periods), but the *SOM* learning and test errors remain on the same level of 3.3. From that it may be concluded that the gravitational clustering algorithm is less robust to the noise in data than the *SOM* algorithm.

The smallest transition point forecasting error - 1.995 periods, was gained by the multi-agent system with the gravitational clustering algorithm, while using the distance measure *MEuclidean*. The test error of two periods may be taken as a relatively high, but, viewing it from the position of the end user, the ability of gaining the additional useful information about the bounds of a PLC phase, makes this error level acceptable.

3. Intelligent Sales Forecasting

Traditional forecasting is mostly based on finding relations in time series, representing changes in economic indexes, system technical parameters etc. It is assumed that relations are mined from a stationary time series and the longer it is, the higher is the probability to find some relations in it (Armstrong et al., 2005). However, in most of real-life tasks the length of a time series is relatively short and mining relations will not bring the desired results. An example of such task is a textile sales analysis and forecasting, which is the main task the proposed system was designed for. As it was stated, clustering textile sales short time series will return sales profiles, but how precise they will be? An additional information can be gained from a descriptive data of each textile product. This sets a task of combinations of several data mining methods - cluster analysis and classification, in order to obtain an additional useful information.

The proposed system is aimed at forecasting textile product sales profiles using only descriptive data, such as price, size, colour, type etc. Subsection 3.1 reveals the structure and conceptual aspects of the proposed system. Figure 8 displays the proposed system.

3.1 Concepts of Sales Forecasting Multi-Agent System

The proposed sales forecasting multi-agent system contains three agents - Data Management Agent, Data Mining Agent and the Sales Forecasting Agent. The Data Management Agent supports receiving data from a data repository and data preparation for further analysis. The Data Mining Agent mines relations from sales data, merging those with product descriptive data and inducing classification rules that are included in the knowledge base. The Sales Forecasting Agent uses the knowledge base to forecast a sales profile for a new product, having only descriptive data available.

3.1.1 Data Management Agent

The Data Management Agent is an agent that is similar to the one described in Subsection 2.1.2. It receives raw data from a data repository, each record containing product descriptive data and one year sales data, and performs data pre-processing - this two steps are performed in the autonomous mode. The Data Management Agent cleans the uninformative profiles like those containing one or two sales facts during all 12 months or have a large number of missing values. The rest of sales profiles that were not excluded, are normalized with *life curve* normalization method (Thomassey & Fioraliso, 2006). Normalizing by the life curve, the new value $x_i^{'}$ of each period of a time series $X = x_1, \ldots, x_i, \ldots, x_n$ is calculated by Equation 13.

$$x_i^{'} = \frac{x_i}{\sum_{j=1}^{n} x_j} \tag{13}$$

The third step of pre-processing is feature selection. During this process a set of the attributes with descriptive data is produced as a an aggregation of available attributes. The objective of this process is to lessen the attributes negative impact to the classification results (Thomassey & Fioraliso, 2006). Attributes can be selected using the information gain or by simple choosing based on user experience. The third - Dataset formation, process is performed reactively by responding to the Data Mining Agent's requests.

3.1.2 Data Mining Agent

The Data Mining Agent is an intelligent agent. The environment it is situated in is the data from the datasets received from the Data Management Agent. The Data Mining Agent performs data clustering and creates classification models, changing the environment into the classification rule sets. The processes in the Data Mining Agent are divided into two main steps - data clustering and the classification rules induction.

The *k-means* algorithm (Tan et al., 2006) was chosen to support the data clustering, though other clustering algorithms can also be applied. The *k-means* is a simple and easy-to-understand algorithm, but it has a sufficient weakness - the number of clusters must be defined before the clustering process begins. Due to that, the clustering process should be repeated several times with different number of clusters and the best number of clusters may be chosen by the smallest clustering error. To lessen the time of searching for the appropriate number of clusters an assumption that the maximal number of clusters should be less or equal to the $\sqrt{n}$, where n is the number of records in the dataset, can be used (Tan et al., 2006). The *k-means* algorithm is executed as follows:

1. The number of clusters m is defined, for each cluster centroid random values are assigned.
2. By using a distance measure (the *Euclidean* distance is used by default) each record in the dataset is assigned to the closest cluster. If none of records changed its cluster, then the clustering process finishes.

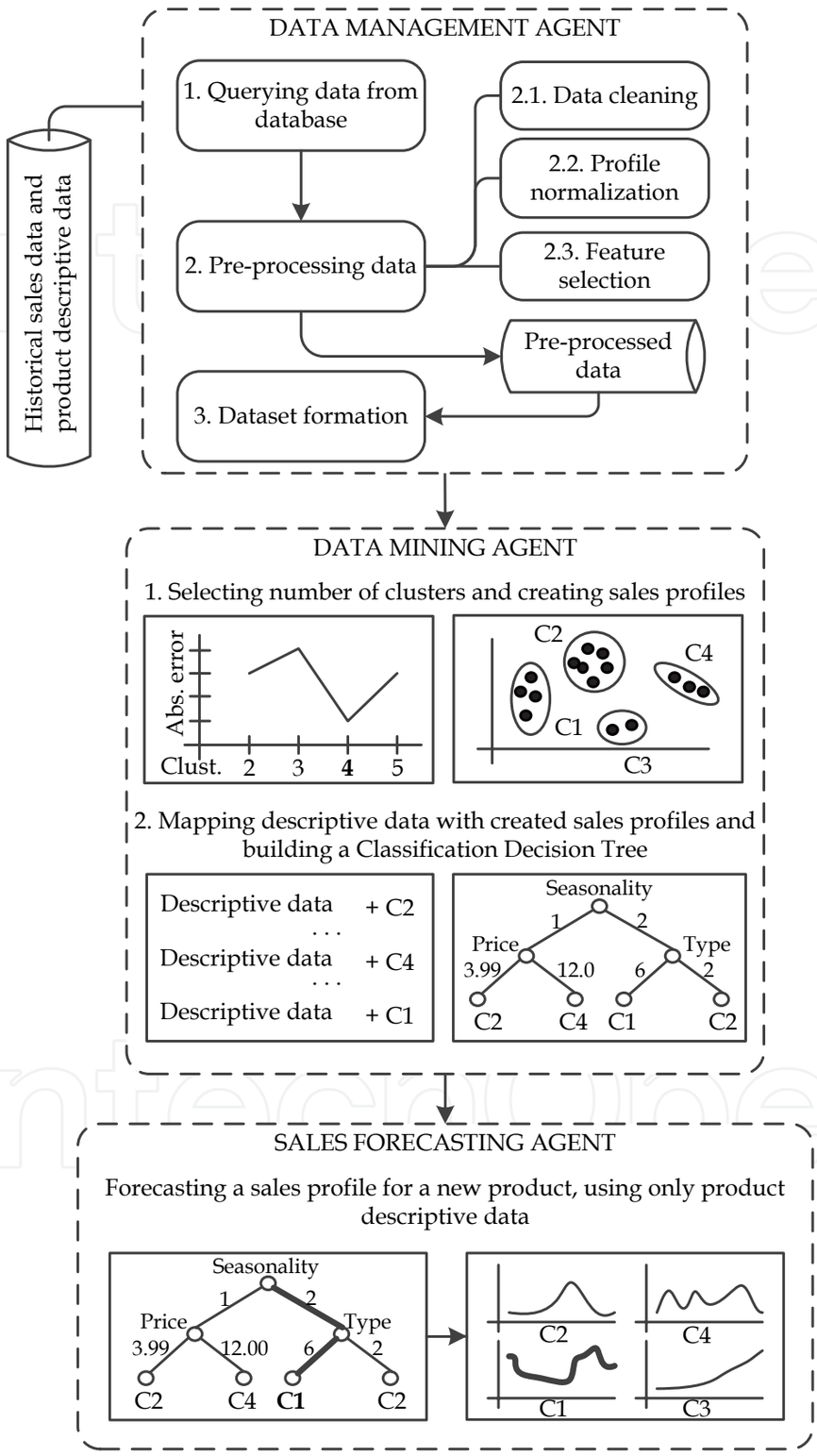


Fig. 8. Multi-Agent System for Sales Forecasting

3. Cluster centroids are recalculated using the records in the appropriate cluster. Then the algorithm returns to Step 2.

Having all data distributed among the defined number of clusters, the clustering error should be calculated in order to show the efficiency of the chosen number of clusters. The clustering error is calculated as a mean absolute error. First, Equation 14 is used to calculate the absolute error AE_i for each cluster c_i. Then the mean absolute error is calculated by Equation 15.

$$AE_i = \frac{1}{n(c_i)} \cdot \sum_{j=1}^{n(c_i)} d_j \ , \tag{14}$$

where $n(c_i)$ is the number of records in the cluster c_i; d_j - the absolute distance between j-th record in the cluster c_i and the centroid of this cluster.

$$MAE = \frac{1}{m} \cdot \sum_{i=1}^{m} AE_i \tag{15}$$

The result of clustering the sales data is the set P, containing a number of sales profiles - cluster centroids, example of which is displayed in Figure 9, where the centroid is marked with a bold line. Each sales profile p is assigned a unique identification number, which will be used in the classification rule set induction step. The data clustering step finishes with the

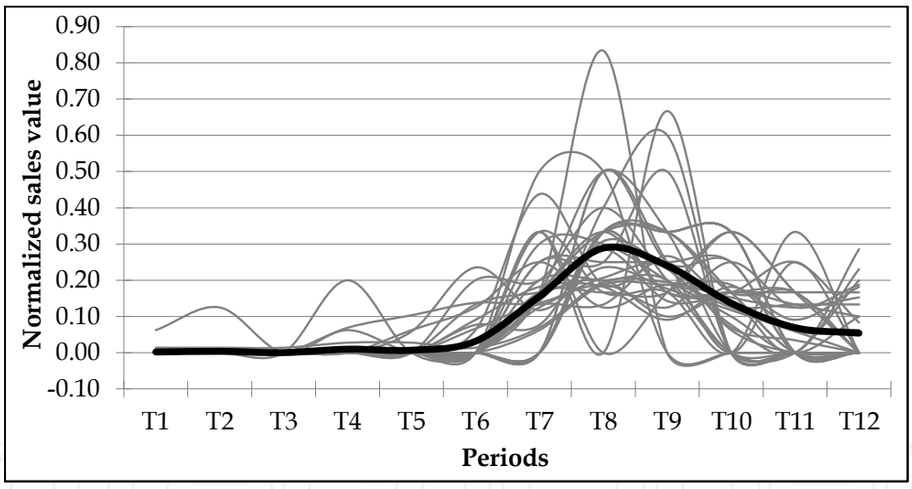


Fig. 9. An example of a cluster and a centroid

process of merging the descriptive data with found sales profiles. During this process each record is merged with an identification number of a closest sales profile. In other words - each record is assigned an appropriate target attribute value.

In order to forecast a sales curve for a product by having only descriptive data, a model of the relations between found sales profiles and descriptive data must be created. Assuming that each sales profile $p \in P$ is a class, the classification methods can be applied for building the desired model. A number of classification methods exist, each having its positive and negative sights. The proposed sales forecasting system applies the inductive decision trees for building the classification model. The decision trees not only create a classification model, which can be represented as a rule set, but also support a graphical tree-like representation of a model, which makes it more understandable for a user.

The Data Mining Agent builds the inductive decision tree using the training dataset, merged with found sales profiles. It uses the C4.5 classification algorithm (Quinlan, 1993). The C4.5 algorithm is able to proceed with discrete and numeric attributes and to handle the missing values. The decision tree built is the model representing the relations between descriptive attributes and the sales profiles. The internal nodes represent the descriptive attributes and leafs are pointing to one of sales profiles found.

3.1.3 Sales Forecasting Agent

The Sales Forecasting Agent is an agent that uses the model (knowledge base) created by the Data Mining Agent. The environment the Sales Forecasting Agent is situated in is the set of new data, containing only descriptive attributes. The environment may include the records from a test set that have the sales profile attached, while the system is in the evaluation stage. The Sales Forecasting Agent changes the environment by forecasting a sales profile for the new data, using the product descriptive data and the model built by the Data Mining Agent. The sales profile forecasting process can be described as a sequence of the next steps:

1. Chose a record X, containing the descriptive data of a new product.
2. By following the structure of the classification tree, created by the Data Mining Agent, find a tree leaf l_j for the record X_i.
3. Take the sales profile p_k, represented by the determined leaf l_j, as a forecasted one for the record X_i.

Figure 10 displays a part of the classification tree and shows the path to the leaf for a record with the following descriptive data: *Kind = 2*, *Type = 1* and *Price = 65.4*.

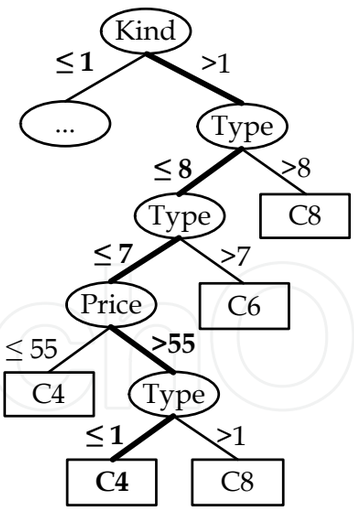


Fig. 10. Example of a classification decision tree

3.2 Experimental results

The proposed sales forecasting system was trained using sales data of a local retailer of textile products, gathered in the year 2005. After all necessary pre-processing steps were finished, the dataset contained 423 records. The 149 sales records from the year 2006 were used for system

testing. As a descriptive data, the following three parameters were chosen: *Kind*, showing either the product is seasonal or not, the *Price* of the product and the *Type*, showing what the product actually is - jeans, t-shirt, bag, shorts, belt etc.
The data clustering was powered by the *k-means* algorithm and the number of clusters was chosen by evaluating clustering error with different cluster number starting with two and continuing till 21, as the square root of 423 records is approximately equal to 20.57. Figure 11 shows the results of clustering error evaluation with different number of clusters. The

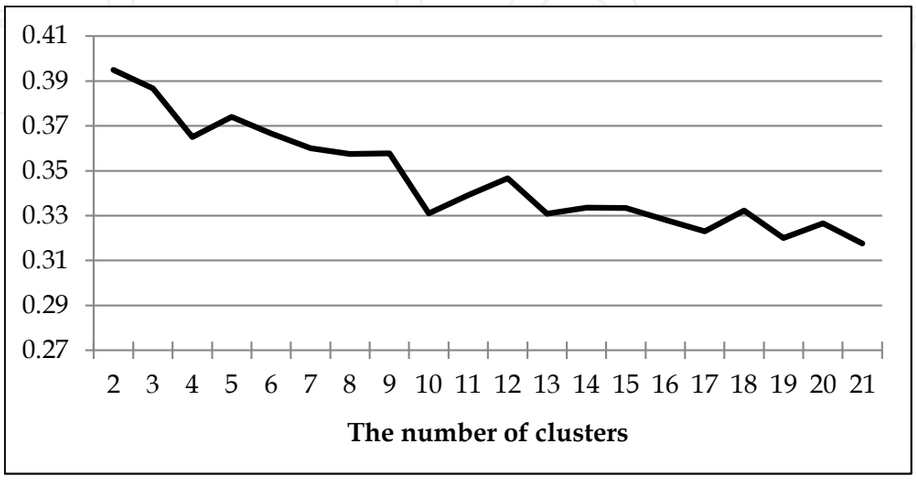


Fig. 11. The clustering error for different number of clusters

first noteworthy improvement in clustering error was while using 10 clusters. For larger numbers of clusters the clustering error was not sufficiently improving, comparing to the result with 10 clusters. Taking this into account, the number of clusters was set to 10 for all further experiments.
For building the classification model, the C4.5 classification algorithm was chosen among other classification methods by comparing six classifiers using such measures as *Root Mean Square Error - RMSE* (see Equation 16) and the *Mean Absolute Error - MAE* (see Equation 17). The *RMSE* and *MAE* were used to measure an error between the forecasted sales profile and the actual sales profile.

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (f_i - x_i)^2} \tag{16}$$

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |f_i - x_i| \ , \tag{17}$$

where n is the number of periods in the forecasted sales profile F and also in the actual sales profile X.
Figure 12 displays the classification tree that was built by the C4.5 classification algorithm. Table 4 shows the results of different classifier evaluation. As can be seen, the C4.5 decision tree algorithm dominates others in both error measures.

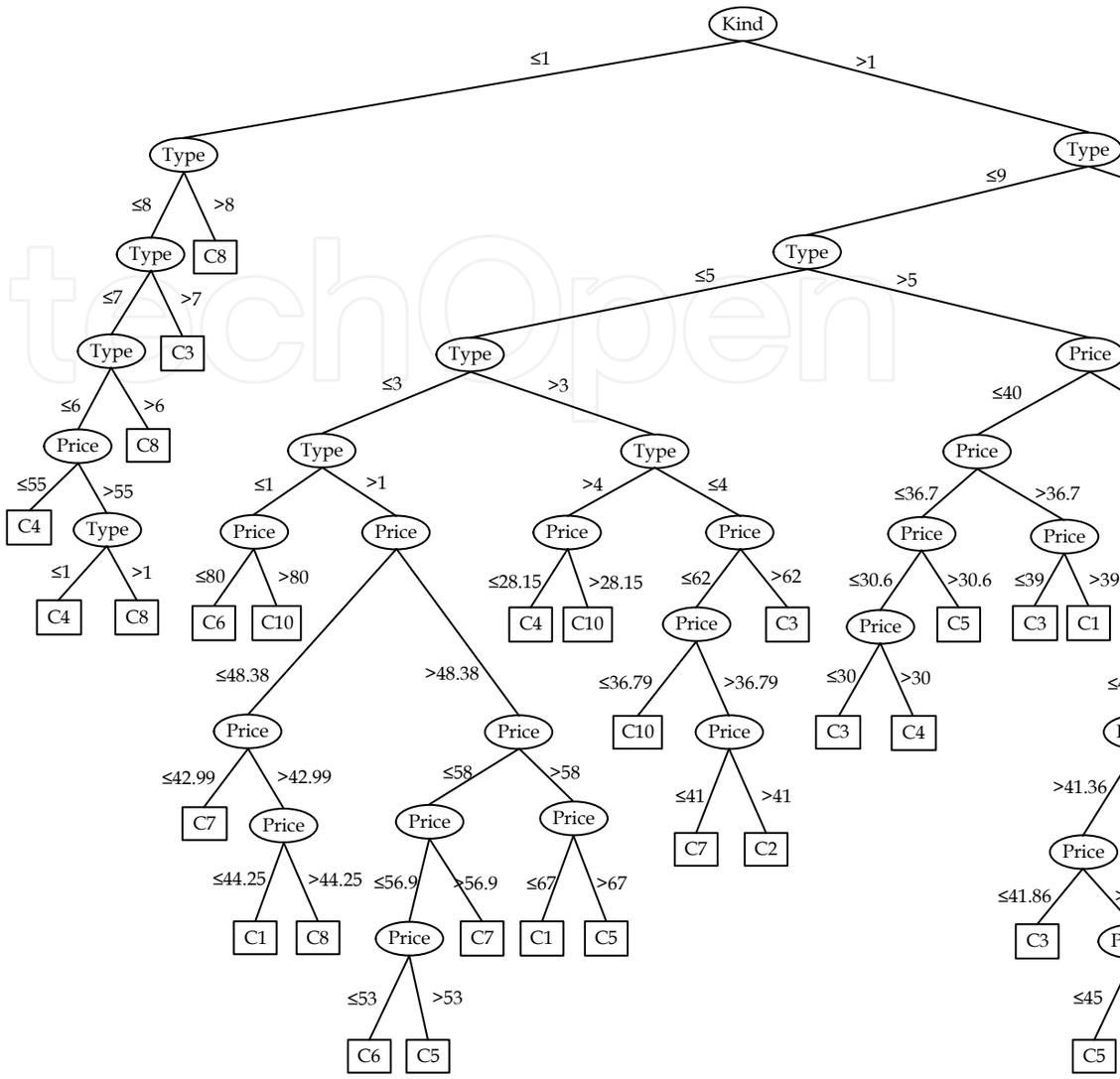


Fig. 12. Decision tree for sales profile forecasting

Error measures	Classifiers					
	ZeroR	*OneR*	*JRip*	*Naive Bayes*	*IBk*	*C4.5*
RMSE	0.295	0.375	0.284	0.282	0.298	**0.264**
MAE	0.174	0.141	0.161	0.156	0.160	**0.129**

Table 4. Error measures for various classification methods

3.3 Results analysis and summary

The results obtained while testing the proposed multi-agent sales forecasting system show that a combination of the intelligent agent technology and the data mining methods - clustering and classification, can be efficiently used for solving such task as the sales forecasting. The system showed good results in clustering the sales data and building the decision tree for forecasting a sales profiles for new products, having only descriptive data available, which is important for retailers while planning the investments for the new stock products. One of the advances of the system is that a user is able to personally view the model - classification decision tree that was used for forecasting to find the relations between the sales profile and the descriptive data, which makes the model more understandable. Using the intelligent agent technology, the system was made modular, which simplifies the implementation of new clustering and classification methods.

4. Conclusions

The modern production and trade management remains a complicated research field and as technology is evolving, becomes even more complicated. This field combines different tasks of forecasting, planning, management etc. In the present research two of them - PLC management and sales forecasting, were discussed and possible solutions proposed. Looking back in time, those tasks were efficiently solved by standard statistical methods and a single user was able to perform that. Today the level of technological evolution, the amount of gathered data and complicated planning objectives require the solutions that are able to lessen the human's load and are capable to perform different data analysis tasks in the autonomous mode. In order to gain the above possibilities, the agent technology and combinations of the intelligent data analysis methods were chosen as the basis. Using the mentioned technologies, the structures of the PLC management support multi-agent system and the multi-agent sales forecasting system were proposed and the concepts of each system were described and discussed. The structures of the proposed systems were intentionally made general in order to demonstrate the advances of the agent technology. The modularity of the proposed systems allows one to adapt individual processes to the specific field of application, without breaking the structure of the system.

Both systems were tested using the real-life data and the results obtained show the effectiveness of the proposed multi-agent systems and the technologies, chosen for creating those systems. The conclusion of the present research is that the intelligent agent technology and other intelligent data analysis technologies can be efficiently applied in tasks connected with supporting the modern production and trade management. The two proposed multi-agent systems give an example of such applications.

5. Acknowledgement

This work has been supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University".

6. References

Aitken, J., Childerhouse, P. & Towill, D. (2003). The impact of product life cycle on supply chain strategy, *International Journal of Production Economics* Vol. 85(2): 127–140.

Armstrong, J. S., Collopy, F. & Yokum, J. T. (2005). Decomposition by causal forces: A procedure for forecasting complex time series, *International Journal of Forecasting* Vol. 21(1): 25–36.

Chang, P.-C., Liu, C.-H. & Fan, C.-Y. (2009). Data clustering and fuzzy neural network for sale forecasting: A case study in printed circuit board industry, *Knowledge-Based Systems* Vol. 22(5): 344–355.

Chien, C.-F., Chen, Y.-J. & Peng, J.-T. (2010). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle, *International Journal of Production Economics* Vol. 128(2): 496–509.

Gomez, J., Dasgupta, D. & Nasraoui, O. (2003). A new gravitational clustering algorithm, *Proceedings of the Third SIAM International Conference on Data Mining*, pp. 83–94.

Keogh, E. J. & Pazzani, M. J. (2001). Derivative dynamic timewarping, *Proceedings of the First SIAM International Conference on Data Mining*, pp. 1–11.

Kohonen, T. (2001). *Self-Organizing Maps*, 3rd edn, Springer-Verlag Berlin Heidelberg.

Ni, Y. & Fan, F. (2011). A two-stage dynamic sales forecasting model for the fashion retail, *Expert Systems with Applications* Vol. 38(3): 1529–1536.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann publishers.

Salvador, S. & Chan, P. (2007). Fastdtw: Toward accurate dynamic time warping in linear time and space, *Intelligent Data Analysis* Vol. 11(5): 561–580.

Tan, P. N., Steibach, M. & Kumar, V. (2006). *Introduction to Data Mining*, Addison-Wesley.

Thomassey, S. & Fioraliso, A. (2006). A hybrid sales forecasting system based on clustering and decision trees, *Decision Support Systems* Vol. 42(1): 408–421.

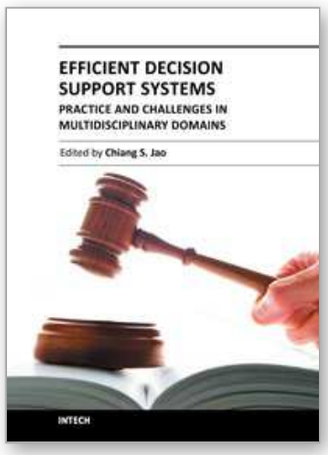
Thomassey, S. & Happiette, M. (2007). A neural clustering and classification system for sales forecasting of new apparel items, *Applied Soft Computing* Vol. 7(4): 1177–1187.

Venkatesan, R. & Kumar, V. (2002). A genetic algorithms approach to growth phase forecasting of wireless subscribers, *International Journal of Forecasting* Vol. 18(4): 625–646.

Weiss, G. (ed.) (1999). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, MIT Press.

Wooldridge, M. (2005). *An Introduction to MultiAgent Systems*, 3rd edn, John Wiley & Sons, Ltd.

Wright, W. E. (1977). Gravitational clustering, *Pattern recognition* Vol. 9: 151–166.



Efficient Decision Support Systems - Practice and Challenges in Multidisciplinary Domains

Edited by Prof. Chiang Jao

ISBN 978-953-307-441-2
Hard cover, 478 pages
Publisher InTech
Published online 06, September, 2011
Published in print edition September, 2011

This series is directed to diverse managerial professionals who are leading the transformation of individual domains by using expert information and domain knowledge to drive decision support systems (DSSs). The series offers a broad range of subjects addressed in specific areas such as health care, business management, banking, agriculture, environmental improvement, natural resource and spatial management, aviation administration, and hybrid applications of information technology aimed to interdisciplinary issues. This book series is composed of three volumes: Volume 1 consists of general concepts and methodology of DSSs; Volume 2 consists of applications of DSSs in the biomedical domain; Volume 3 consists of hybrid applications of DSSs in multidisciplinary domains. The book is shaped decision support strategies in the new infrastructure that assists the readers in full use of the creative technology to manipulate input data and to transform information into useful decisions for decision makers.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen